

**MÓDULO DE SIMULACIÓN PARA TRANSFERENCIA DE DATOS EN
COMUNICACIÓN MÓVILES**

P R E S E N T A:

GUSTAVO ADOLFO LÓPEZ GÓMEZ

**CORPORACION UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERIA
TECNOLOGIA EN REDES DE COMPUTADORES Y SEGURIDAD
INFORMATICA**

BOGOTÁ, COLOMBIA 2010

MÓDULO DE SIMULACIÓN PARA TRANSFERENCIA DE DATOS EN COMUNICACIÓN MÓVILES

GUSTAVO ADOLFO LÓPEZ GÓMEZ

TESIS DE GRADO PARA OPTAR AL TÍTULO
DE TECNÓLOGO EN REDES DE COMPUTADORES Y SEGURIDAD INFORMÁTICA

Director de Proyecto:

EDGAR MAURICIO RUIZ
INGENIERO DE SISTEMAS

CORPORACION UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERIA
TECNOLOGIA EN REDES DE COMPUTADORES Y SEGURIDAD INFORMATICA
BOGOTA DC.
2010

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Jurado

A los __ días del mes _____ del año _____. Bogotá DC.

DEDICATORIA

A mis padres.

Beatriz y Gustavo por su apoyo incondicional la formación y educación.

A mis amigos más cercanos.

Porque siempre están ahí apoyándome sin importar las circunstancias.

AGRADECIMIENTOS

Yo Gustavo Adolfo López Gómez, autor de este proyecto de grado deseo expresar mi agradecimiento a las siguientes personas que colaboraron durante el proceso de elaboración y culminación de este trabajo:

A Dios y a mi familia que me apoyaron incondicionalmente durante todo mi proceso de formación y en los momentos más difíciles de mi vida, y sin mencionar los mejores momentos.

A mi asesor Mauricio Ruiz por el apoyo, colaboración y dedicación durante el desarrollo de este proyecto.

A un sinfín de amigos que he conocido en el transcurso de mi vida ya que me han enseñado muchas cosas y hemos compartido experiencias de vida, todo eso me ha formado el carácter y me hizo ser la persona que hoy soy.

Muchas Gracias,
Gustavo Adolfo López Gómez

CONTENIDO

Pág.

CAPÍTULO I: PROBLEMA

1. PLANTEAMIENTO DEL PROBLEMA	15
1.1 IDENTIFICACIÓN DEL PROBLEMA	15
1.1.1 Formulación Del Problema	16
1.2 JUSTIFICACIÓN	16
1.3 OBJETIVOS.....	18
1.3.1 Objetivo General	18
1.3.2 Objetivos Específicos	18
1.4 ALCANCE	18
1.5 LIMITACIONES	19
1.6 HIPÓTESIS	19
1.6.1 General	19
1.6.2 De Trabajo	20
1.7 ESTADO DEL ARTE	20
1.7.1 Antecedentes De La Investigación	20

CAPÍTULO II: MARCO REFERENCIAL

2. ¿QUÉ ES ANDROID?.....	21
2.1 USUARIO TÉCNICO.	21
2.1.1 Como programar en Android.....	21
2.1.2 Arquitectura (Android A.) ^[3]	22
2.1.2.1 Nucleó – Linux	22
2.1.2.2 Librerías.....	23
2.1.2.3 Runtime de Android.....	23
2.1.2.4 Framework de Aplicaciones	23
2.1.2.5 Aplicaciones.....	24
2.1.3 Estructura de una Aplicación	24
2.1.3.1 Activity.....	25

2.1.3.2 IntentReceiver	25
2.1.3.3 Service	25
2.1.3.4 ContentProvider	25
2.1.4 Jerarquía Visual de Android	26
2.1.4.1 Views	26
2.1.4.2 Viewgroups.....	26
2.2. USUARIO FINAL	27
2.2.1 Celulares con Android	27
2.3 QUE APORTA ANDROID	28
2.4 LICENCIA ANDROID.....	29

CAPÍTULO III: METODOLOGIA

3. METODOLOGÍA	30
3.1 DISEÑO DE LA INVESTIGACIÓN	30
3.1.1 Metas.....	31
3.1.2 Funciones del Sistema	31
3.2 ANÁLISIS DE REQUERIMIENTOS	32
3.2.1 Funcionales	32
3.2.2 No Funcionales.....	33
3.3 MODELO DE DESARROLLO DE SOFTWARE	34
3.3.1 Metodología Incremental Evolutiva.....	34

CAPÍTULO IV: DISEÑO

4. ARQUITECTURA.....	35
4.1 INTERFACES.....	35
4.2 MODELO ENTIDAD RELACIÓN	36
4.3 DIAGRAMA DE ACTIVIDADES	37
4.4 DIAGRAMAS DE SECUENCIA	37
4.5. DIAGRAMA DE CLASES.....	41
4.5.1 Servidor	41
4.5.2 Cliente	42

4.6 DOCUMENTACIÓN	43
4.6.1 Diagrama De Casos De Uso	43
4.6.1.1 Especificación de casos de uso	44
4.6.2 Pruebas	47
4.6.2.1 Caja Blanca.....	47
4.6.2.2 Caja Negra.....	49

CAPÍTULO V: CRONOGRAMA

5. CRONOGRAMA.....	51
5.1 CRONOGRAMA DE ACTIVIDADES	51
5.2 PRESUPUESTO	52

CAPÍTULO VI: CONCLUSIONES

6. CONCLUSIONES	53
6.1 RECOMENDACIONES	54

BIBLIOGRAFIA

7. BIBLIOGRAFÍA.....	55
----------------------	----

LISTA DE FIGURAS

Pág.

Figura 1. Arquitectura de Android.	22
Figura 2. Estructura de una Aplicación Android.....	24
Figura 3. Celulares Android.	27
Figura 4. Metodología Incremental Evolutiva.	34
Figura 5. Inicio Figura 6. Registro Figura 7. Panel de Control	36
Figura 8. Modelo Entidad Relación	36
Figura 9. Diagrama de Flujo de Proceso	37
Figura 10. Diagrama de Secuencia de Registrar	38
Figura 11. Diagrama de Secuencia de Autenticar	38
Figura 12. Diagrama de Secuencia de Enviar	39
Figura 13. Diagrama de Secuencia de Cerrar.....	39
Figura 14. Diagrama de Secuencia de Chatear	40
Figura 15. Diagrama de Secuencia de Registros	40
Figura 16. Diagrama de Clases - Servidor.	41
Figura 17. Diagrama de Clases - Cliente	42
Figura 18. Diagrama de Casos de Uso	43
Figura 19. Prueba Caja Blanca – DDMS	48

LISTA DE TABLAS

	Pág.
Tabla 1. Celulares Compatibles con Android.....	28
Tabla 2. Requerimiento Funcional 1.	32
Tabla 3. Requerimiento Funcional 2.	32
Tabla 4. Requerimiento Funcional 3.	32
Tabla 5. Requerimiento Funcional 4.	33
Tabla 6. Requerimiento Funcional 5.	33
Tabla 7. Requerimiento No Funcional 1.	33
Tabla 8. Especificación CU1.	44
Tabla 9. Especificación CU2.	45
Tabla 10. Especificación CU3.	45
Tabla 11. Especificación CU4.	46
Tabla 12. Especificación CU5.	46
Tabla 13. Especificación CU14.	47
Tabla 14. Resultados Pruebas P1.	49
Tabla 15. Resultados Pruebas P2.	49
Tabla 16. Resultados Pruebas P3.	50
Tabla 17. Resultados Pruebas P4.	50
Tabla 18. Actividades.	51
Tabla 19. Presupuesto.....	52

LISTA DE ANEXOS

GLOSARIO

ANDROID: Es un sistema operativo para móviles al igual que Windows es un sistema operativo para ordenadores.

API LEVEL: Es un valor entero único que identifica el marco de revisión de la API, asociada con la versión de la plataforma Android, es decir, se encarga de revisar todas las funciones y procedimientos que tendrán las diferentes versiones de la plataforma Android.

API: Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

AVD: Android virtual devices, es una aplicación en donde se pueden configurar las opciones del emulador y además de esto escoger la API LEVEL del emulador de Android.

DALVIK: Es una maquina virtual que utiliza la plataforma para dispositivos móviles Android.

JAVA: Es un lenguaje de programación.

JDK: Es un software que provee herramientas de desarrollo para la creación de programas en java.

PLATAFORMA: Es un concepto abstracto, que se puede definir como los cimientos en donde se puede llevar a cabo un desarrollo de software.

PLUGIN: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

SDK: Es un conjunto de herramientas y programas necesarios para que los desarrolladores lleven a cabo su trabajo.

SMARTPHONES: Un smartphone es un teléfono móvil que incorpora características de una computadora personal.

STACK: Es una lista o estructura de datos en la cual el modo de acceso a sus elementos es de tipo LIFO (Ultimo en entrar, primero en salir).

RESUMEN

Este proyecto es llevado a cabo a partir de la iniciativa de describir por medio de la ingeniería de software lo que es el proceso de investigación, desarrollo y construcción de un aplicativo de software para las tecnologías móviles de celulares con sistema operativo Android.

El principal interés de este es dar a conocer el nuevo sistema operativo móvil denominado Android, realizando pruebas de su comportamiento en cuanto a comunicación, por medio de los simuladores Android Virtual Device (AVD), e incentivar a los estudiantes de tecnologías de la información a continuar con este tipo de proyectos.

Además, es necesario crear una serie de talleres y guías explicativas paso a paso en donde los estudiantes o cualquier persona interesada en el tema, puedan acceder a la documentación y paradigma de programación para sistemas Android en lenguaje Java, facilitando el entendimiento para que puedan desarrollar, implementar y modificar productos que utilicen este tipo de tecnología.

ABSTRACT

This project is conducted from the describe initiative through which software engineering is the process of research, development and construction of a software application for mobile cellular technologies operating system Android.

The main interest of this is to introduce the new mobile operating system called Android, by testing their behavior in communication, through the simulators Android Virtual Device (AVD), and encourage students to information technology continue this type of project.

Furthermore, it is necessary to create a series of workshops and explanatory step by step guides in which students or anyone interested in the subject, have access to documentation and programming paradigm for Android systems in Java, providing the understanding to enable them to develop, implement and modify products using this technology.

Capítulo I: PROBLEMA

"La formulación de un problema es, a menudo, más importante que su solución". (Albert Einstein)

1. PLANTEAMIENTO DEL PROBLEMA

1.1 IDENTIFICACIÓN DEL PROBLEMA

Hoy en día, es posible ver como en cortos periodos de tiempo, nacen nuevas tecnologías móviles y es nuestra obligación el conocerlas para poder utilizarlas con el mayor provecho posible. Actualmente las comunicaciones por medio de los dispositivos móviles son utilizadas por una gran variedad de personas y en un mayor porcentaje por la juventud de 15 a 25 años de edad, sin importar su sexo, religión o ideología política.

Como resultado de la acogida que tiene las tecnologías móviles, se puede observar el incremento del uso de estas para comunicarse de diferentes maneras implementando y adoptando diferentes tipos de medios como lo son wifi, bluetooth, infrarrojo, etc., así ofreciendo a sus usuarios las posibilidades de interactuar con un sinnúmero de servicios ofrecidos por terceros.

Las tendencias actuales, indican que el sistema operativo Android, marcara la pauta en el mercado de tecnologías móviles. Por este motivo, las grandes empresas desarrolladoras de software, están orientando sus esfuerzos en la construcción de aplicativos eficientes y de calidad para este nuevo sistema operativo.

1.1.1 Formulación Del Problema

¿Cómo se podría generar un nuevo software de simulación que permita mostrar el proceso de comunicación de datos para las tecnologías móviles basados en el sistema operativo Android?

¿Por quienes podría ser utilizado este aplicativo?

1.2 JUSTIFICACIÓN

Los constantes descubrimientos tecnológicos hacen evolucionar inevitablemente la mentalidad de las personas, es por esto que basándose en las nuevas tecnologías móviles que se imponen en el mercado se ha optado por desarrollar una aplicación para el sistema operativo Android, que es un sistema operativo móvil muy nuevo pero a pesar de esto ha causado un gran impacto y tiene una gran acogida por los usuarios.

Android no es sólo para programadores y sería un error pensar eso. Android es un Sistema Operativo igual que cualquier otro con la diferencia que detrás de él hay miles de personas haciendo programas para él y esto es la primera vez que sucede en los móviles, hasta ahora nadie había dado facilidades para realizar programas para móviles, sí existían pero no con este potencial ya que al ser código libre los programadores pueden hacer cualquier cosa, y eso es muy beneficioso para todos los usuarios. Pero un usuario no programador también puede participar y ayudar, solo con acceder a todos los programas de la gente, tanto los que lo hayan hecho gratis para todo el mundo como los de pago, opinando sobre los programas de la gente, decirles qué posibles mejoras pueden hacer, diciendo ideas que se te ocurran o incluso simplemente probando las aplicaciones y diciéndoles si falla algo o no.

Teniendo Android en un móvil o aparato cabe la posibilidad de obtener miles de programas, porque no hay una empresa haciendo programas, si no miles de personas y cada una con ideas diferentes.

Por ser Android un sistema operativo muy reciente, son pocas las personas en Colombia que tienen el interés y el conocimiento sobre este sistema para desarrollar un software específico. Es por esto, que es el momento de ser pioneros investigando y dando a conocer este sistema desarrollando una aplicación eficiente, aprovechando que su estructura de programación es basada sobre lenguaje Java, el cual es el principal lenguaje de programación inducido a los estudiantes de Redes y Seguridad Informática de la Universidad Minuto de Dios. Al desarrollar este proyecto se podrá obtener como beneficios para los estudiantes de la facultad de Ingeniería de Uniminuto, el dar a conocer este nuevo sistema operativo móvil, explicando sus ventajas y desventajas al compararlos con otros sistemas logrando así de esta manera una nueva temática a la cual ellos pueden interesarse, permitiendo mejores críticas y dejando abierto el tema para futuras mejoras.

Este proyecto parte por el deseo de investigar y dar a conocer el sistema operativo Android por medio de una aplicación que se implemente en las tecnologías móviles de celulares y explore sus recursos, por esto se presenta una propuesta totalmente innovadora denominada "REDPHONE", que busca entregar al usuario final una comodidad y flexibilidad en cuanto a comunicación. Implementado las técnicas, habilidades y aptitudes para garantizar una aplicación estable, productiva, rica en recursos y fácil de utilizar, que sea de interfaz agradable, que funcione y se pruebe en los simuladores Android Virtual Device (AVD) para su igual comportamiento en los dispositivos físicos reales.

1.3 OBJETIVOS

1.3.1 Objetivo General

Desarrollar un aplicativo de software denominado “REDPHONE” que tiene como función principal la transferencia de datos, para ser implementado en tecnologías móviles que utilizan sistema operativo Android.

1.3.2 Objetivos Específicos

- ✓ Desarrollar las clases y métodos que permitan la comunicación y transferencia de datos en Java, utilizando las librerías del sistema operativo Android.
- ✓ Crear las interfaces necesarias en lenguaje XML que faciliten la interacción con el usuario.
- ✓ Crear una base de datos en SQLite que cumpla con las características necesarias para el correcto funcionamiento de la aplicación.
- ✓ Construir los manuales técnico, y de usuario del Software.
- ✓ Diseñar el esquema lógico de conectividad de usuarios del aplicativo.

1.4 ALCANCE

El alcance de este proyecto consiste en el diseño y planteamiento de una aplicación denominada REDPHONE. El producto de software a entregar consta del modulo de transferencia de datos para dispositivos móviles, el cual fue diseñado específicamente para llevar a cabo unas determinadas tareas preestablecidas de comunicación, como lo es compartir información por medio de mensajería, a través de conexión TCP/IP.

La documentación comprende el manual de usuario, manual técnico y manual del administrador, con características tipo guía, las cuales poseen ejemplos claros y explicados para su respectivo destinatario.

El software y documentación tendrá licencia de distribución de software libre con código abierto (Open Source), con el fin de que terceros puedan realizar nuevos módulos y enriquecer la aplicación REDPHONE.

1.5 LIMITACIONES

Las pruebas se ejecutan sobre los simuladores Android Virtual Device (AVD), que es una aplicación multiplataforma que virtualiza las principales características del celular “Google phone Dream / G1”, logrando así el mismo comportamiento de este en el PC.

El simuladores AVD no fue diseñado para funcionar en red, por lo que no es posible comunicar dos simuladores que están en diferentes maquinas, pero sin embargo se pueden comunicar varios simuladores en una misma máquina. Por lo tanto las pruebas se realizaran en una maquina con dos dispositivos AVD comunicados por Telnet.

Del comportamiento obtenido como resultado de las pruebas, será el punto para juzgar y concluir el funcionamiento del mismo en los dispositivos físicos reales.

1.6 HIPÓTESIS

1.6.1 General

La creación de un sistema de comunicación para las tecnologías móviles, facilitara la interacción de los estudiantes con nuevas tecnologías y fomentara la integración entre ellos mismos, sin importar la carrera que estén estudiando dentro de la universidad.

1.6.2 De Trabajo

Los estudiantes de tecnologías de la información tendrán un enorme apoyo en cuanto a documentación, referencias y métodos a utilizar en sus aplicaciones, resolviendo las dudas que puedan poseer.

1.7 ESTADO DEL ARTE

1.7.1 Antecedentes De La Investigación

Pensando en celulares Android nace como alternativa a los magnates actuales del mercado que son los siguientes: Microsoft, con la evolución de Windows Mobile que han sido implementados en dispositivos Handheld PC, Pocket PC, teléfonos inteligentes o Smartphone. Palm, con su sistema Palm OS, quien fue líder de mercado hasta el año 2004, pero después de varios errores en el desarrollo de su sistema, perdió terreno y ahora es uno de los menos utilizados. Nokia con Symbian, que hasta hace pocos años ha empezado a cosechar terreno, pero con muy buenas ventajas en cuanto a ahorro de energía y memoria. Y Apple con su Apple OS, que solo lo vemos en un modelo con dos versiones (iPhone) logra tener una cantidad impresionante de aparatos vendidos.

En cuanto a proyecto de investigación y desarrollo en sistema operativo Android en Colombia cabe destacar el trabajo realizado por dos estudiantes Juan David y Estiven en la Universidad de Medellín al producir una aplicación denominada Track Incomes and Expenses (TIE)^[1], que nació a partir de un estudio realizado en mayo y agosto de 2008 en Boston, Estados Unidos por Juan David que es uno de sus desarrolladores. TIE es básicamente un administrador de finanzas personales (ingresos y egresos), el cual supero las expectativas ya que después de 4 días de haber sido lanzado al mercado había superado las 1000 descargas por usuarios de todo el mundo.

Capitulo II: MARCO TEORICO

"El conocimiento teórico es un tesoro cuya llave es su práctica." (Thomas Fuller)

2. ¿QUÉ ES ANDROID?

2.1 USUARIO TÉCNICO.

Android es un stack de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones clave. Google está otorgando un vistazo al SDK de Android, que provee de herramientas y APIs necesarios para comenzar a desarrollar aplicaciones en la plataforma Android, utilizando el lenguaje de programación Java.

2.1.1 Como programar en Android

Para empezar a programar en Android es importante tener conocimientos de Java, así como de Programación Orientada a Objetos. Además de Java se utiliza XML para la parte visual, otro lenguaje descriptivo muy fácil de utilizar y que aporta sencillez a la hora de programar en Android.

Se puede programar en cualquier plataforma Windows, MAC o Linux. Para ello ahí que descargar el Android SDK, que consta de todas las librerías Java que utiliza Android, el emulador para poder probar las aplicaciones (Ver Manual Técnico).

2.1.2 Arquitectura (Android)^[2]

La arquitectura interna de la plataforma Android está básicamente conformada por:

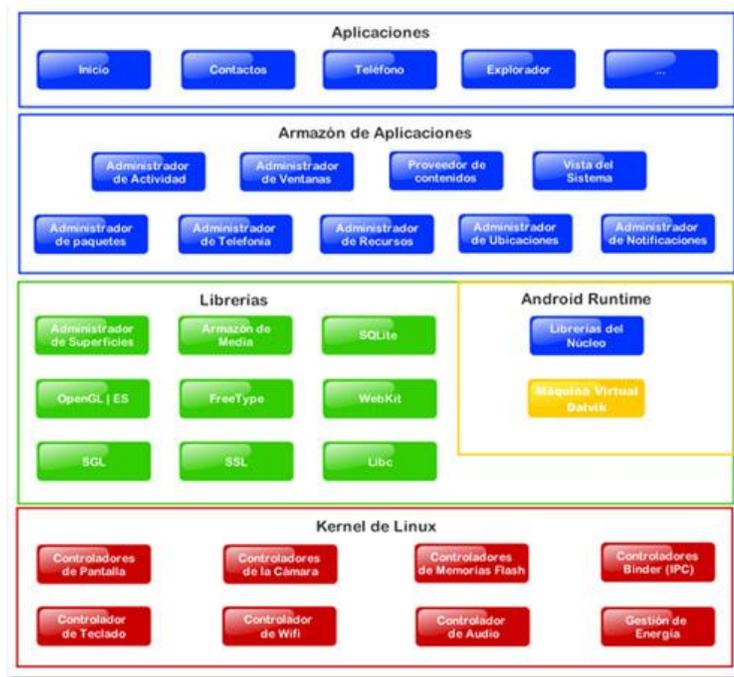


Figura 1. Arquitectura de Android.

2.1.2.1 Nucleó - Linux

Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del stack de software.

2.1.2.2 Librerías

Android incluye un set de librerías C/C++ usadas por varios componentes del sistema Android. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library (implementación librería C standard), librerías de medios, librerías de gráficos, 3d, SQLite, entre otras.

2.1.2.3 Runtime de Android

Android incluye un set de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros, y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

2.1.2.4 Framework de Aplicaciones

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar el reutilizar componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario.

2.1.2.5 Aplicaciones

Es la última capa, la capa de aplicación es donde se ejecutan las aplicaciones que se han programado usando las API. En esta capa encontramos las aplicaciones básicas para que el móvil funcione y además todas las que se instalen en el dispositivo.

2.1.3 Estructura de una Aplicación

En la estructura de una aplicación Android nos encontramos con la integración de diferentes componentes. Toda aplicación hace uso de las APIs para que los componentes que realizan cada tarea puedan ser acezados, manipulados o reemplazados en cualquier momento sin interrumpir el proceso para así asegurar la mayor confiabilidad posible. Los principales componentes de una aplicación son:

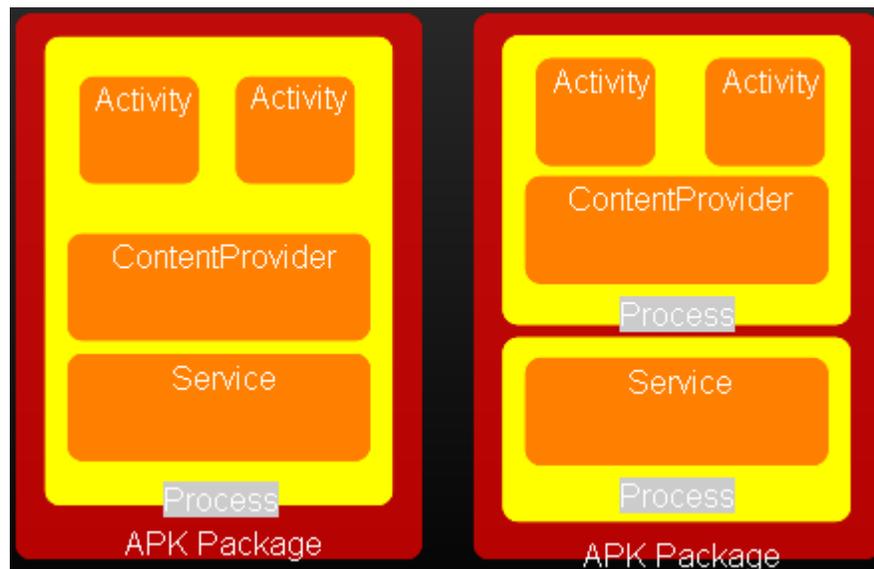


Figura 2. Estructura de una Aplicación Android.

2.1.3.1 Activity

Una actividad es el bloque de construcción de una interfaz usuario. Una equivalencia de una actividad podría decirse de manera analógica, una actividad es una ventana de escritorio, también es posible tener una actividad sin interfaz usuario.

2.1.3.2 IntentReceiver

Son mensajes del sistema, que se ejecutan y corren en el interior del dispositivo, para notificar a las aplicaciones de los eventos que van ocurriendo, como por ejemplo un cambio de estado del hardware que podría ser una tarjeta de memoria insertada, también los datos entrantes que podrían ser mensajes. No solo responde a estas “intents”, además de las que viene por default se pueden agregar nuevas que dependan de alguna aplicación específica.

2.1.3.3 Service

Servicios que están diseñados para mantenerse en funcionamiento independientes de cualquier actividad. Actividades proveedores del contenido y receptores de “intents”, que por lo general son de corta duración y se pueden apagar en cualquier momento.

2.1.3.4 ContentProvider

Son proveedores de contenido que proporcionan un nivel de atracción de los datos que están almacenados en el dispositivo que están disponibles en múltiples aplicaciones. Lo que esto quiere decir es que establece una capa que le permite a las diferentes tipos de aplicaciones compartir los datos con independencia de almacenamiento.

2.1.4 Jerarquía Visual de Android

La principal clase de Android es `activity`, un objeto de la clase `android.app.Activity`. Una actividad hace multitud de cosas, pero por ella misma no se presenta nada en la pantalla. Para conseguir que aparezca algo en la pantalla es necesario diseñar el Interfaz Usuario (UI), con `views` y `viewgroups`, que son las clases que se usan para crear la interfaz entre el usuario y la plataforma Android.

2.1.4.1 Views

Un `view` es un objeto cuya clase es `android.view.View`. Es una estructura de datos cuyas propiedades contienen los datos de la capa y la información específica del área rectangular de la pantalla. Una `view` tiene: `layout`, `drawing`, `focus change`, `scrolling`, etc.

La clase `view` es útil como clase base para los `widgets`, que son unas subclases ya implementadas dibujan los elementos en la pantalla. Los `widgets` contienen sus propias medidas, pero se pueden usar para construir una interfaz más rápidamente. La lista de `widgets` usables incluyen: `Text`, `EditText`, `InputMethod`, `MovementMethod`, `Button`, `RadioButton`, `CheckBox`, `ScrollView`, etc.

2.1.4.2 Viewgroups

Un `viewgroup` es un objeto de la clase `android.view.ViewGroup`, como su propio nombre indica, un `viewgroup` es un objeto especial de `view` cuya función es contener y controlar la lista de `views` y de otros `viewgroups`. Los `viewgroups` permiten añadir estructuras a la interfaz y acumular complejos elementos en la pantalla que son implementados por una sola entidad.

La clase viewgroup es útil como base de la clase layouts, que son subclases implementadas que proveen los tipos más comunes de los layouts de pantalla. Los layouts proporcionan una manera de construir una estructura para una lista de views.

2.2. USUARIO FINAL

Android es un Sistema Operativo, es un Sistema Operativo que ha sido diseñado (por Google) para implementarse sobre todo en aparatos pequeños como móviles, PDAs, etc. Todos esos aparatos llevan un Sistema Operativo detrás de ellos, es decir, Android se puede integrar en cualquier dispositivo (con algunos mínimos requerimientos).

Existen varias versiones de Android: 1.1, 1.5 (Cupcake), 1.6 (Donut), 2.0/2.1 (Éclair). Hay que tener en cuenta que cada versión tiene sus propias características, tienen funciones diferentes y soportan diferentes tipos de aplicaciones (una aplicación de 2.1 no funcionará en 1.6).

2.2.1 Celulares con Android



Figura 3. Celulares Android.

Últimamente se ha visto el incremento en el interés de los fabricantes de modelos de celulares en utilizar sistema operativo Android o ser compatible a este sistema. Dentro de los principales modelos de celulares actuales en el mercado podemos encontrar los siguientes modelos:

Nexus One	Eris / Desire	Behold II	Spica / Galaxy Lite / i5700	Hero
Passion / Bravo	One	DROID / Milestone	Pulse	Galaxy / i7500
GW880	Streak	Archos Phone	Mini 3i	Magic / myTouch 3G
Xperia X10a	Calgary / Motus	CLIQ / DEXT	Sprint Hero	Dream / G1
Eve / GW620	Liquid / A1	O1	Tattoo	Moment

Tabla 1. Celulares Compatibles con Android.

Las especificaciones técnicas y nuevos diseños de estos y otros modelos, se pueden encontrar en: <http://www.androphones.com/all-android-phones.php>. Para este año se han anunciado 12 nuevos modelos, según la publicación de The New York Times: http://www.nytimes.com/2009/10/26/technology/26android.html?_r=2&ref=technology

2.3 QUE APORTA ANDROID

Lo primero que cabe resaltar es el multitask, a pesar de que esto gasta más batería o recurso, los dispositivos móviles tienen que permitir hacer varias cosas a la vez, no hay que escoger entre escuchar música o navegar por internet, entre uno o lo otro lo mejor es todo a la vez y esto me lo ofrece Android.

En cuanto a personalizar se trata, se sabe que cada persona lo hace a su gusto, con sistemas operativos Apple o Microsoft, permiten solo aplicar temas, que básicamente solo cambian los colores e iconos, pero siempre están los accesos directos en el mismo lugar, con Android se puede colocar todos los accesos en cualquier sitio del escritorio, destacando también que trae varios escritorios lo que amplía más las opciones. Pero no solo se cuenta con la modificación interna, por ejemplo si se quiere un iPhone y no agrada

su diseño físico no se puede hacer nada, en cambio con Android solo es buscar un dispositivo que tenga el mejor aspecto y que se adapte a las necesidades entre todos los dispositivos que actualmente hay en el mercado y los que vienen.

Actualmente Android tiene una comunidad enorme y en crecimiento, aportando miles de aplicaciones, juegos, ROMs y modificaciones, lo que hace que un usuario de Android se sienta más cercano a su teléfono, sabiendo que lo puedo modificar mucho más que un usuario Apple o Microsoft.

Sincronización con cuentas Google, el correo y diferentes cuentas se actualizan en todo momento con la web (Internet), también con posibilidades de sincronización con redes sociales.

En la lucha de los gigantes, una empresa como Apple es experta en desarrollo de hardware, y Microsoft es experta en software, lo que hace que el producto de cada empresa tienda a tener defectos según su fabricante. Google es experto en desarrollo de software y al ofrecer código abierto se puede contar con la experiencia en hardware de otras empresas como HTC, Sony Ericsson, Samsung, Motorola; permitiendo que Google se centre mas en el software, que fue lo logrado por la Open Handset Alliance.

2.4 LICENCIA ANDROID

Android se encuentra dividido en 2 licencias: General Public License v2 (GPLv2) para su kernel cuyo núcleo es una versión optimizada del kernel Linux y Apache License v2 para sus aplicaciones (Android Market). Google con la Open Handset Alliance plantean a Android con una licencia de Apache Software License (ASL), a diferencia de GPL, ASL permite a las operadoras y programadores modificar a su gusto la plataforma Android, añadir nuevas funcionalidades, nuevas aplicaciones y no tener que compartir el nuevo código permitiendo restringir ese uso por terceros.

Capítulo III: METODOLOGÍA

"Lo maravilloso de aprender algo, es que nadie puede arrebatárnoslo." (B. B. King)

3. METODOLOGÍA

3.1 DISEÑO DE LA INVESTIGACIÓN

El estudio de la investigación es del tipo diseño de campo, que permite la recolección de datos desde la fuente y sin alterar las condiciones existentes, por lo que se puede afirmar que esta investigación es de carácter no experimental y está basado en las siguientes nociones:

Explorativo, ya que no existen muchos precedentes de trabajos o investigación relacionados con el tema. Lo que conlleva a desarrollar un arduo trabajo cualitativo flexible para aprovechar la información reunida, y poder construir el modelo en el cual se desarrollará el sistema.

Descriptivo, porque revela las características principales del sistema operativo Android, que no poseen los demás de su misma línea. Logrando así una tabla comparativa-descriptiva para analizar su factibilidad y usabilidad.

3.1.1 Metas

La principal meta de este proyecto es desarrollar e implementar un aplicativo compatible con el sistema operativo Android, para abrir las puertas a los nuevos conocimientos y tecnologías, encaminando a que toda persona que tenga vínculo alguno con las Tecnologías de la Información y Comunicación (TIC), basándose en la filosofía de código abierto para que puedan conocerlo, entenderlo y mejorarlo con el fin de retroalimentar la aplicación.

3.1.2 Funciones del Sistema

La aplicación REDPHONE tendrá la arquitectura cliente/servidor, en donde ambos tienen características en común pero sin embargo el servidor tiene privilegios sobre el cliente, permitiéndole ejecutar acciones remotas. REDPHONE tiene una característica principal y es que el usuario no es el que determina el rol entre cliente y servidor, sino que es el ejecutable de la aplicación el encargado de determinar este rol, por lo que cuenta con dos ejecutables uno denominado REDPHONE-SERVER y otro REDPHONE-CLIENT.

Para evitar conflictos en la comunicación, se establece que solo se pueden comunicar un usuario tipo servidor con un usuario tipo cliente, si dos usuarios son tipo servidor o cliente no se podrán comunicar entre sí.

La base de datos en la cual se llevan los registros, se aloja en el dispositivo donde se ejecute el usuario tipo server, por lo tanto si un usuario inicia en otro dispositivo no contará con la misma base de datos.

3.2 ANÁLISIS DE REQUERIMIENTOS

3.2.1 Funcionales

Listado de requerimientos funcionales (R1):

Nombre: R1. 1. Registrar un usuario.

Resumen:	Cuando la persona quiere ingresar a la aplicación lo primero que tiene que hacer es registrarse en la misma.
Entradas:	Los datos ingresados por el usuario en el formulario de registro.
Resultados:	Un nuevo usuario listo para utilizar la aplicación.

Tabla 2. Requerimiento Funcional 1.

Nombre: R1. 2. Eliminar usuario.

Resumen:	Un usuario no se puede eliminar a sí mismo, se contacta con el usuario servidor y le comunica que no desea continuar en la aplicación.
Entradas:	Nick del usuario a eliminar.
Resultados:	El usuario ya no pertenece a la aplicación.

Tabla 3. Requerimiento Funcional 2.

Nombre: R1. 3. Identificación de usuario.

Resumen:	Luego de registrarse el usuario puede iniciar sesión para empezar a disfrutar de los beneficios de la aplicación.
Entradas:	Nick del usuario y su respectiva contraseña.
Resultados:	Permite enviar y recibir archivos y/o mensajes.

Tabla 4. Requerimiento Funcional 3.

Nombre:	R1. 4. Transferencia de Datos TCP y UDP
Resumen:	Da la opción al usuario de enviar y recibir mensajes a otro usuario con la característica que puede o no estar conectado el usuario destinatario.
Entradas:	Mensajes a enviar pertenecientes al usuario.
Resultados:	Mensajes enviados al usuario específico.

Tabla 5. Requerimiento Funcional 4.

Nombre:	R1. 5. Asignación de Usuarios a redes lógicas.
Resumen:	La aplicación automáticamente según los datos suministrador por el usuario al momento de registro, lo asigna a su respectiva subred.
Entradas:	Nombre de la carrera a la cual pertenece el usuario.
Resultados:	Sectoriza a el usuario con los demás de su misma carrera.

Tabla 6. Requerimiento Funcional 5.

3.2.2 No Funcionales

Lista de requerimientos no funcionales (R2):

Nombre:	R1. 1. Registros
Resumen:	La aplicación automáticamente crea registros de las autentificaciones y las comunicaciones exitosas para determinar quién y cuándo utilizo la aplicación.
Entradas:	Inicios de sesión, peticiones de transferencia de archivos y/o mensajes.
Resultados:	Crea log's que permiten demostrar si el archivo y/o mensaje fue enviado.

Tabla 7. Requerimiento No Funcional 1.

3.3 MODELO DE DESARROLLO DE SOFTWARE

Se escoge el modelo de desarrollo de software incremental evolutivo para el diseño de la aplicación REDPHONE porque en esta metodología el producto final no se entrega de una vez, sino que se divide y se entrega por incrementos, lo que quiere decir esto es que se le otorga prioridad a los requisitos funcionales principales establecidos, pero sin embargo los incrementos del producto tiene que ser funcionales no dependientes, lo que corresponde a nuestro proyecto que entrega el modulo de comunicación de la aplicación.

Las ventajas de trabajar de esta manera, es que no tenemos que esperar a que el aplicativo este 100% terminado, y al entregar los primeros incrementos se pueden detectar posteriores requerimientos, logrando abarcar los servicios de sistema con la más alta prioridad. De esta manera se puede manejar el aplicativo por versiones, teniendo una compatibilidad de reemplazo de un una versión antigua a una posterior satisfaciendo las nuevas necesidades.

3.3.1 Metodología Incremental Evolutiva

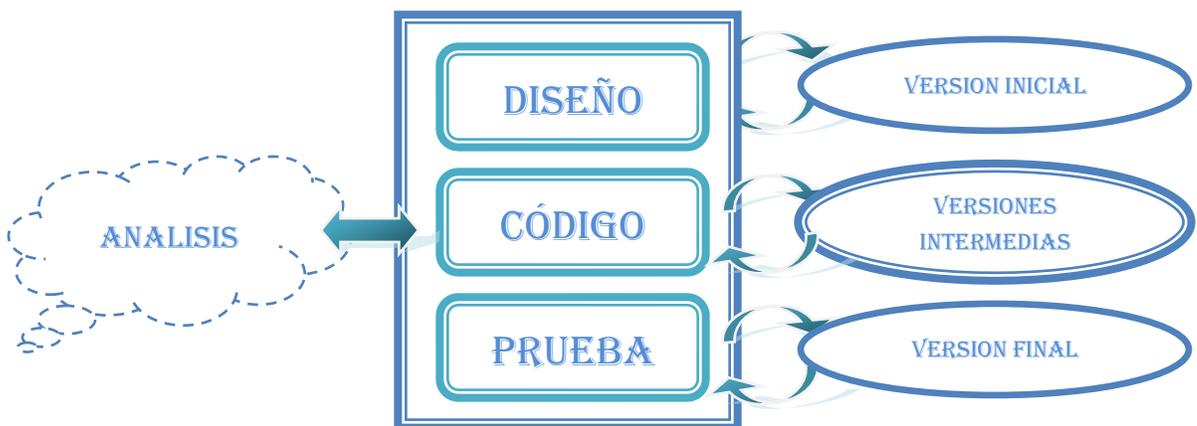


Figura 4. Metodología Incremental Evolutiva.

Capítulo IV: DISEÑO

Hay algo más importante que la lógica: la imaginación. (Alfred Hitchcock)

4. ARQUITECTURA

La arquitectura del software alude a la “*estructura global del software y las formas en que la estructura proporciona la integridad conceptual de un sistema*” (Pressman, 2002), en otras palabras es el centro del sistema donde se encuentra alojado todo lo que es código.

4.1 INTERFACES

El principal uso de las siguientes interfaces es proporcionar al usuario un entorno visual sencillo para permitir la comunicación con el sistema. Pensadas y diseñadas para facilitar la comprensión, aprendizaje y uso de las mismas. (Ver anexo Manual Usuario).

Lo primero que se le muestra al usuario es pantalla de bienvenida (Figura 5) donde el usuario puede identificarse o registrarse. Si el usuario opta por registrarse se envía la interfaz de formulario (Figura 6), donde llena los datos personales y si cumple con los requisitos es agregado a la base de datos. Si se identifica se envía al panel de control (Figura 7), en el cual puede acceder a toda nuestra aplicación.

Desde el panel de control se puede enviar archivos (Figura 10) o mensajes (Figura 11), también posee la opción de ver los archivos compartidos por otros usuarios (Figura 12).

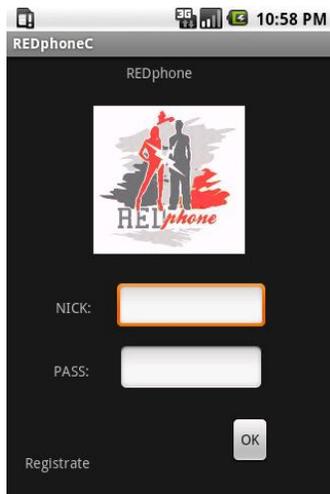


Figura 5. Inicio

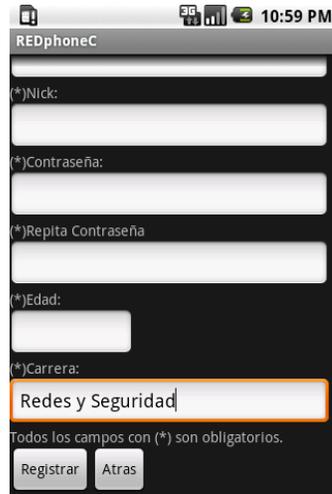


Figura 6. Registro

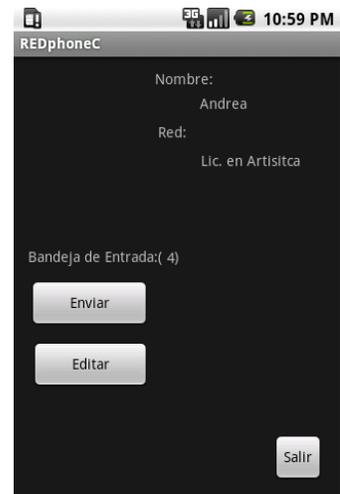


Figura 7. Panel de Control

4.2 MODELO ENTIDAD RELACIÓN



Figura 8. Modelo Entidad Relación

4.3 DIAGRAMA DE ACTIVIDADES

El siguiente diagrama representa un flujo paso a paso de la aplicación REDPHONE, determinando las principales operaciones del sistema:

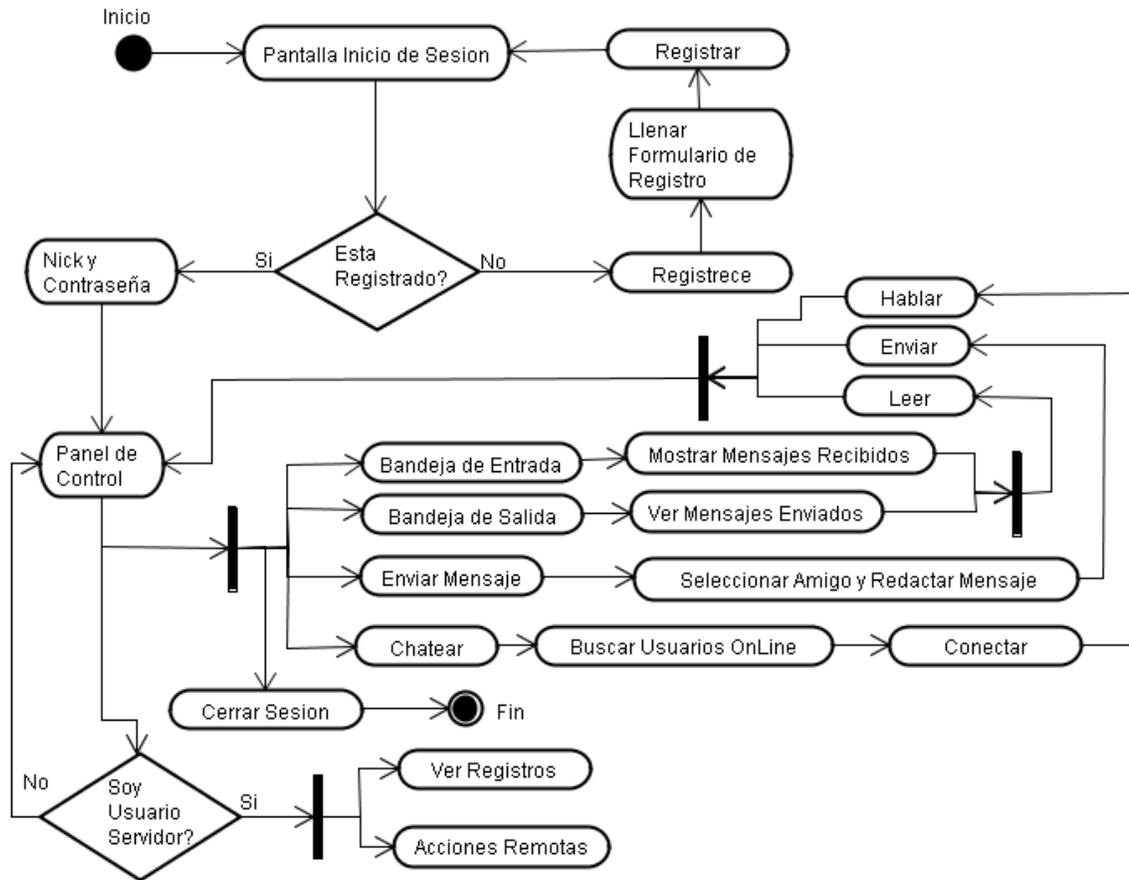


Figura 9. Diagrama de Flujo de Proceso

4.4 DIAGRAMAS DE SECUENCIA

Representa una línea del tiempo donde muestra la integración de un conjunto de objetos con la aplicación, implementando un escenario y mensajes intercambiados entre ellos.

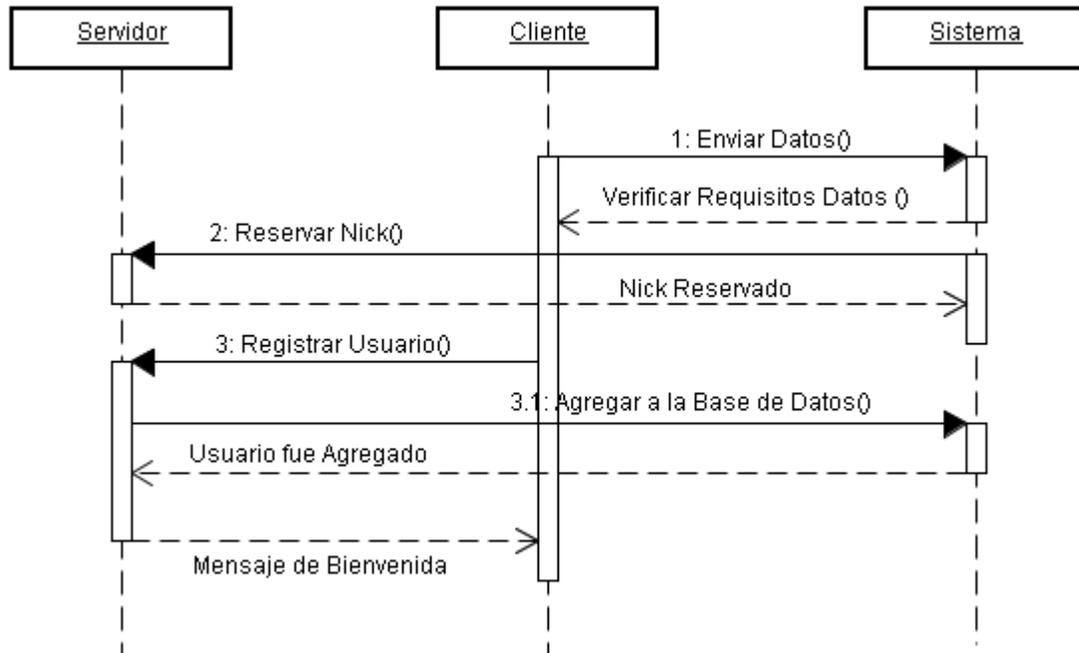


Figura 10. Diagrama de Secuencia de Registrar

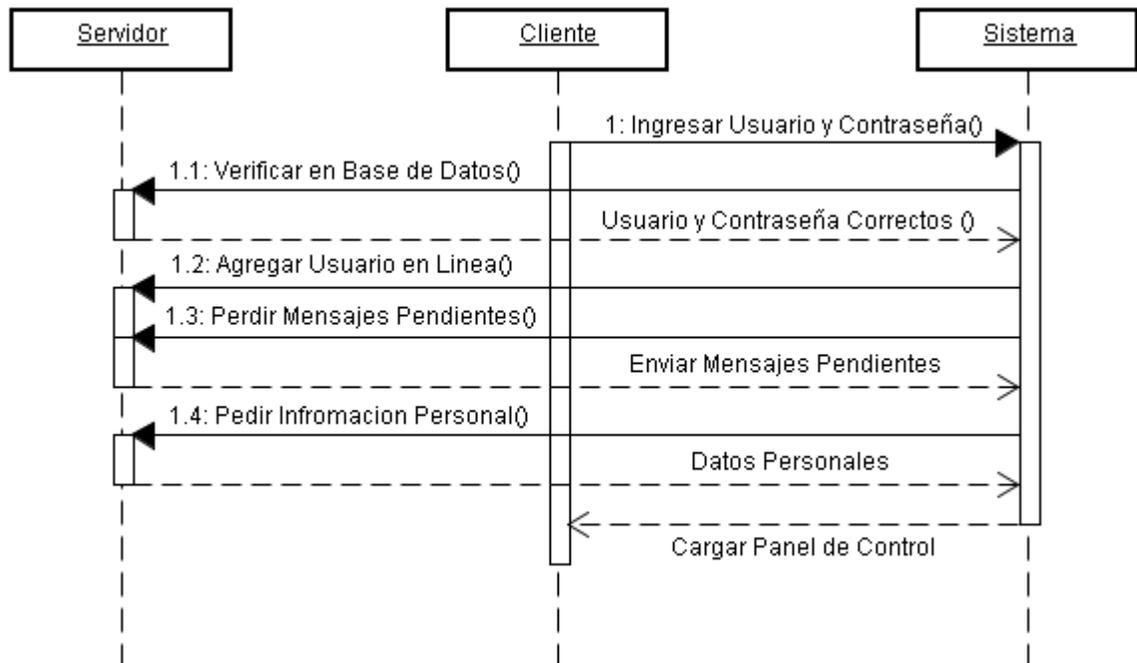


Figura 11. Diagrama de Secuencia de Autenticar

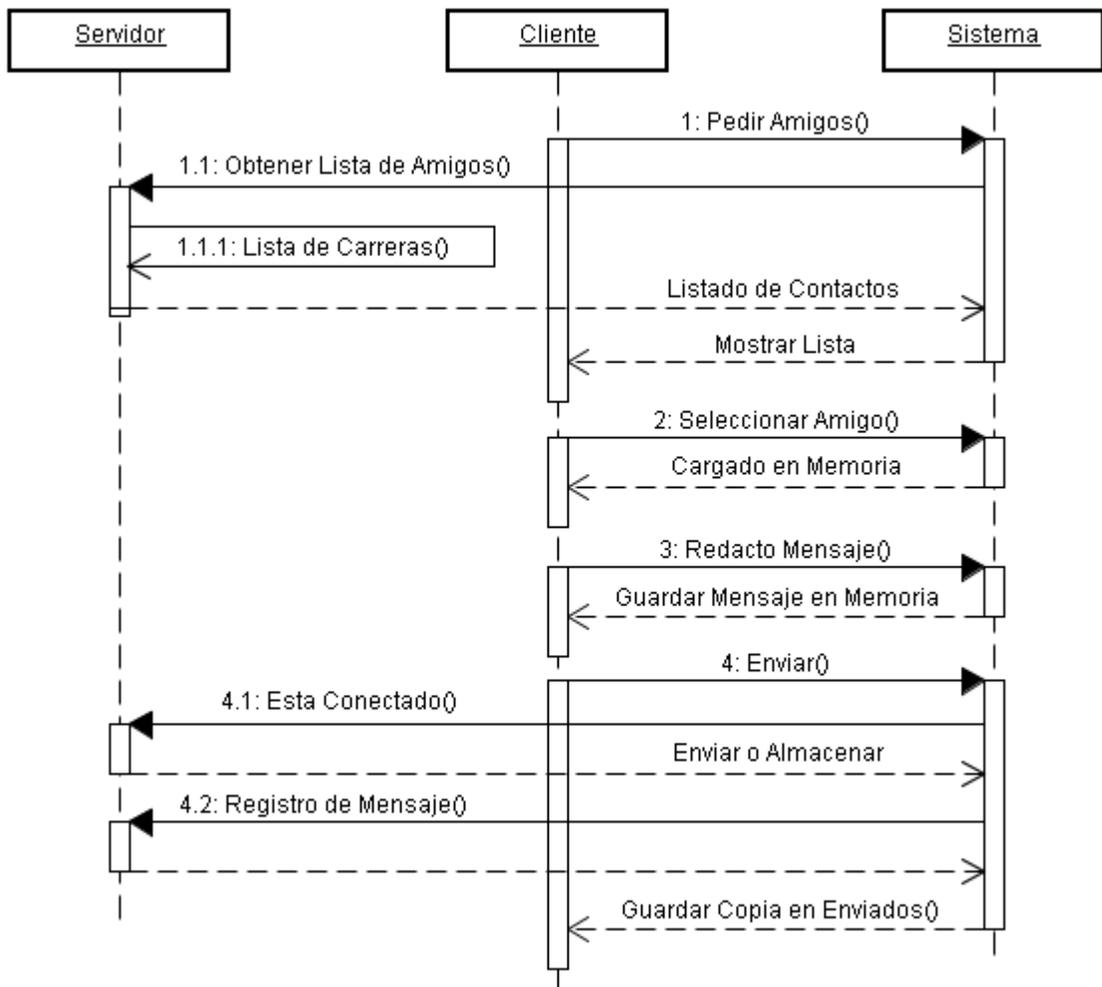


Figura 12. Diagrama de Secuencia de Enviar

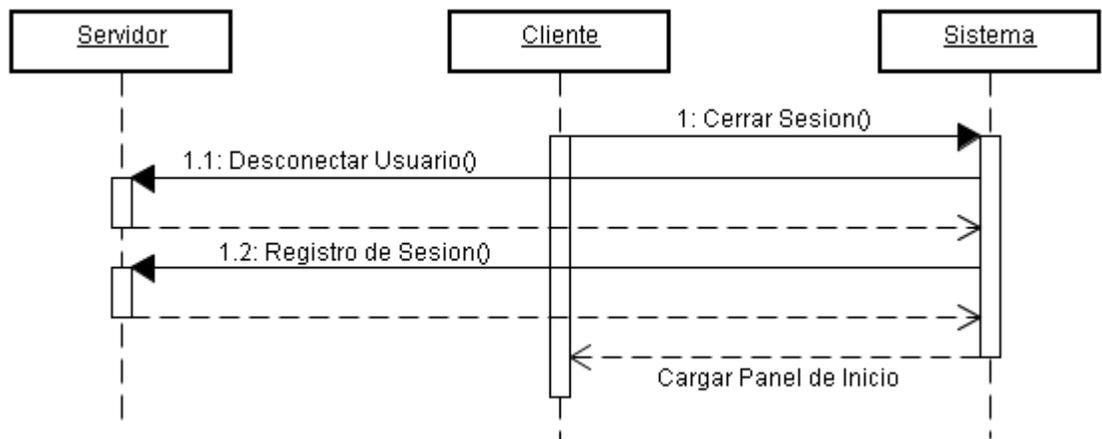


Figura 13. Diagrama de Secuencia de Cerrar

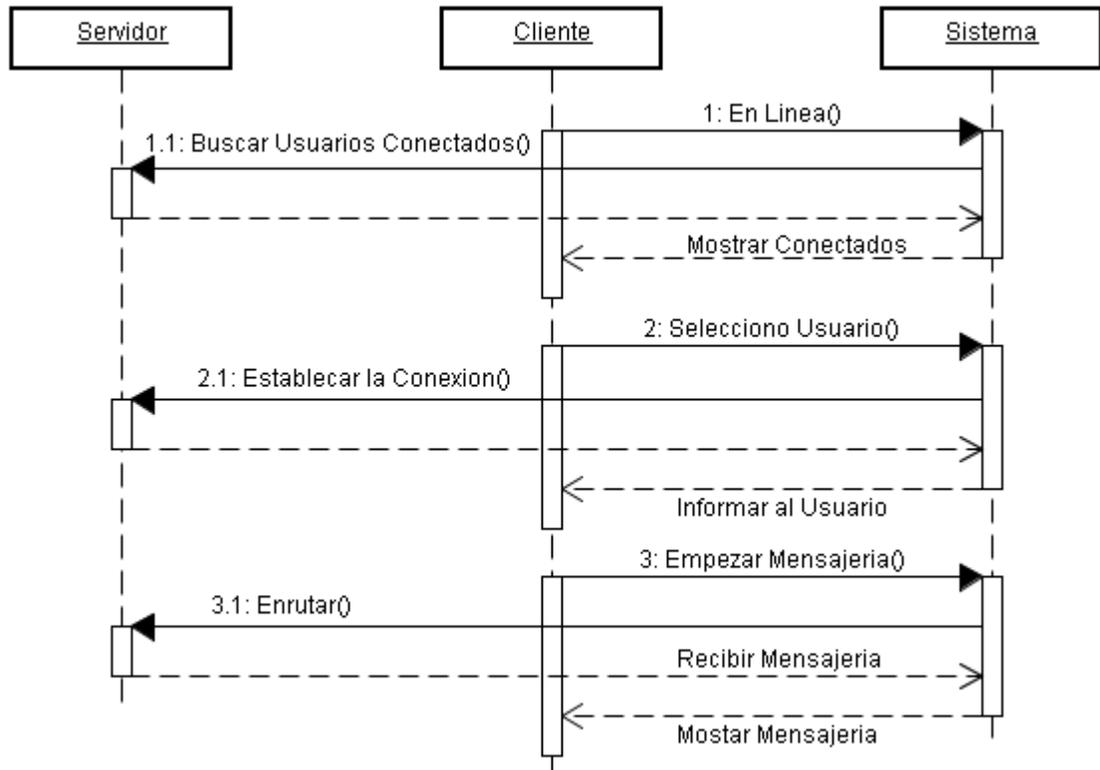


Figura 14. Diagrama de Secuencia de Chatear

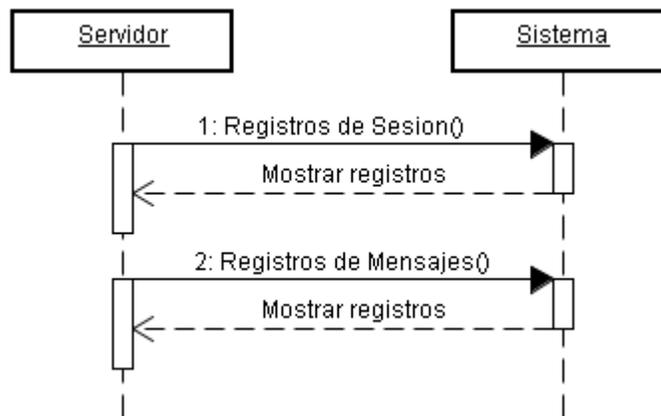


Figura 15. Diagrama de Secuencia de Registros

4.5. DIAGRAMA DE CLASES

Los diagramas de clases básicamente describen la estructura de un sistema, mostrando sus clases, atributos y relaciones entre ellos (Ver API's de Redphone).

4.5.1 Servidor

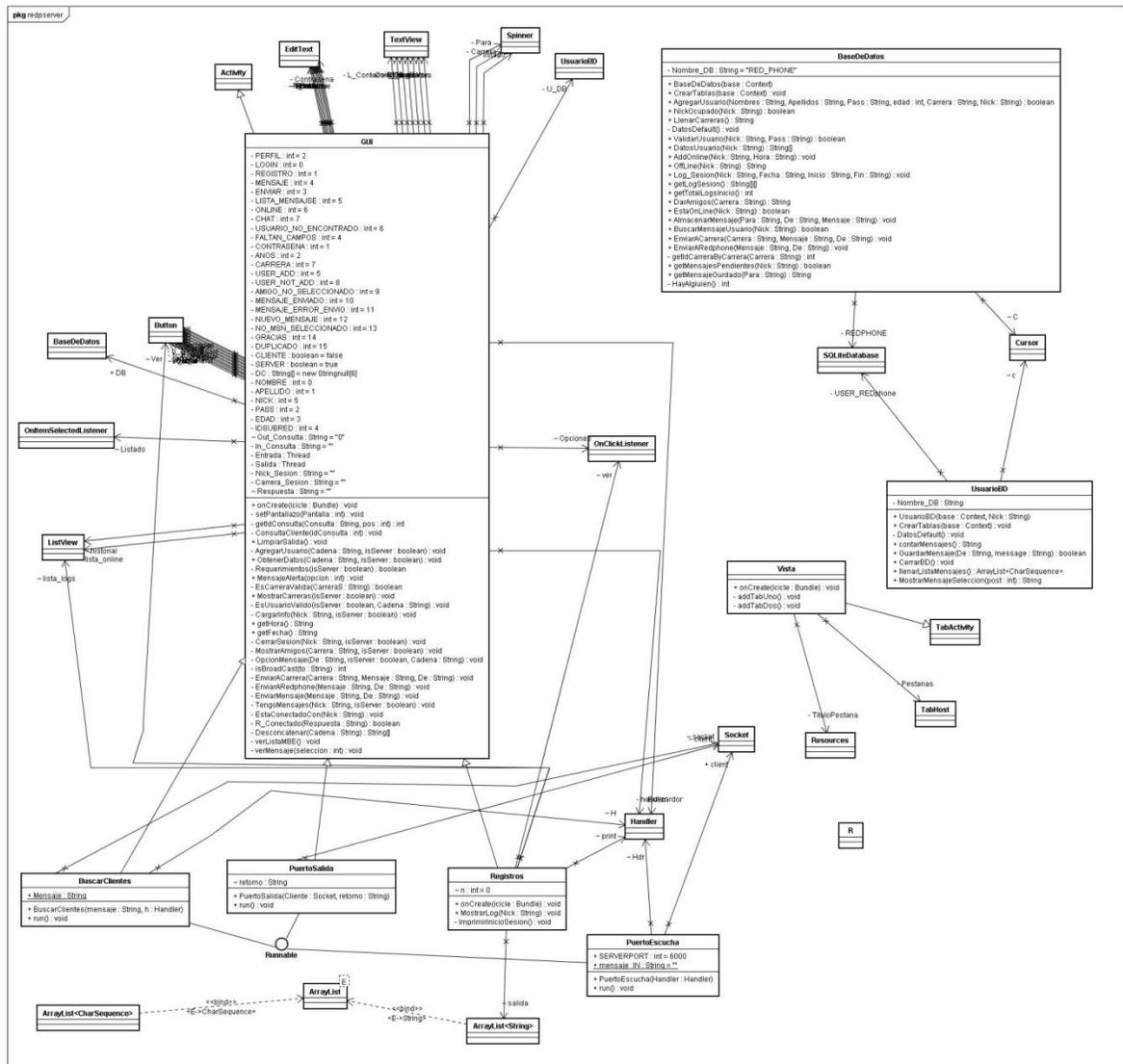


Figura 16. Diagrama de Clases - Servidor.

4.5.2 Cliente

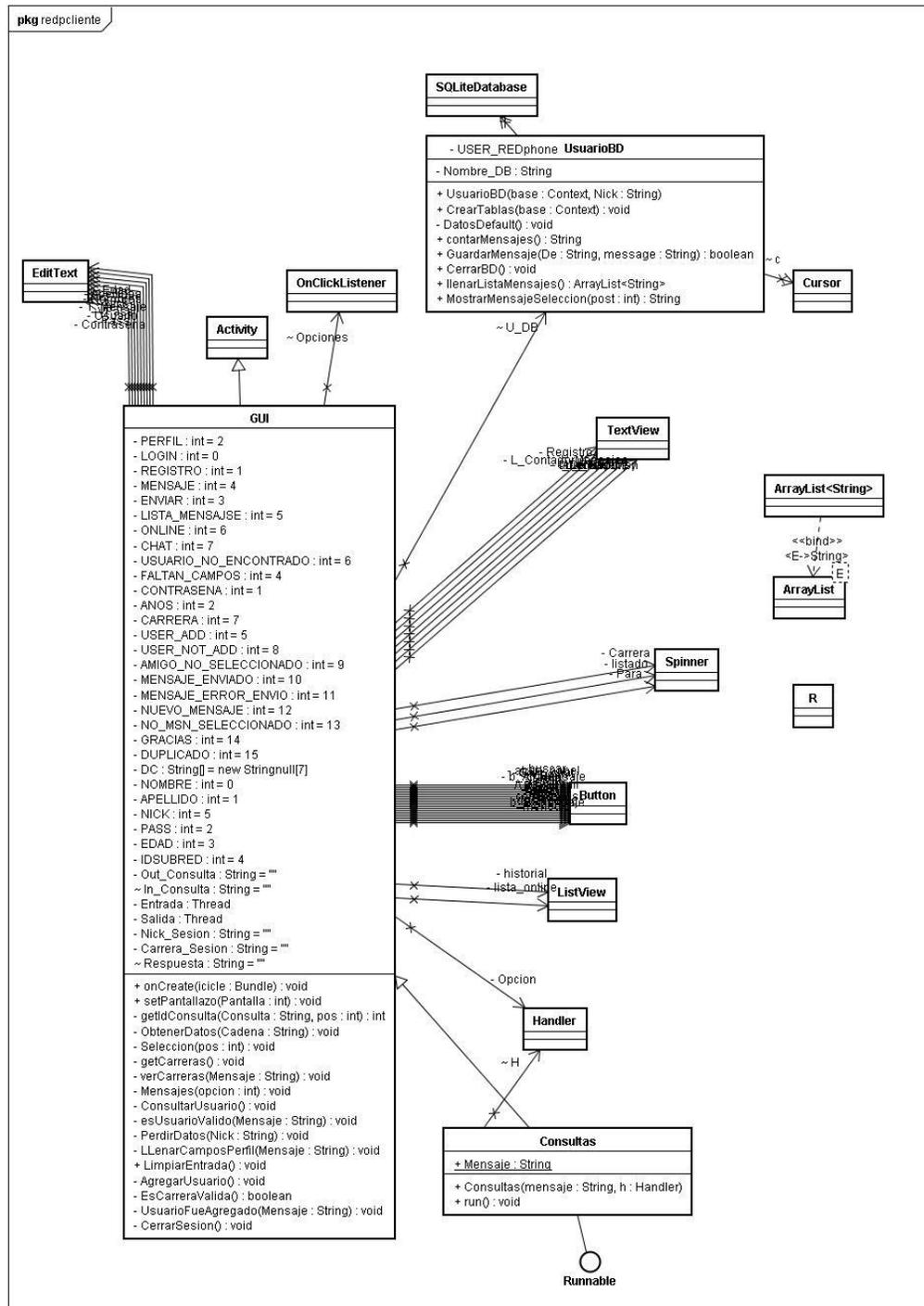


Figura 17. Diagrama de Clases - Cliente

4.6 DOCUMENTACIÓN

La documentación proporciona el fundamento para un buen desarrollo, y lo que es más importante proporciona guías para la tarea de mantenimiento del software (Ver Anexos Manuales).

4.6.1 Diagrama De Casos De Uso

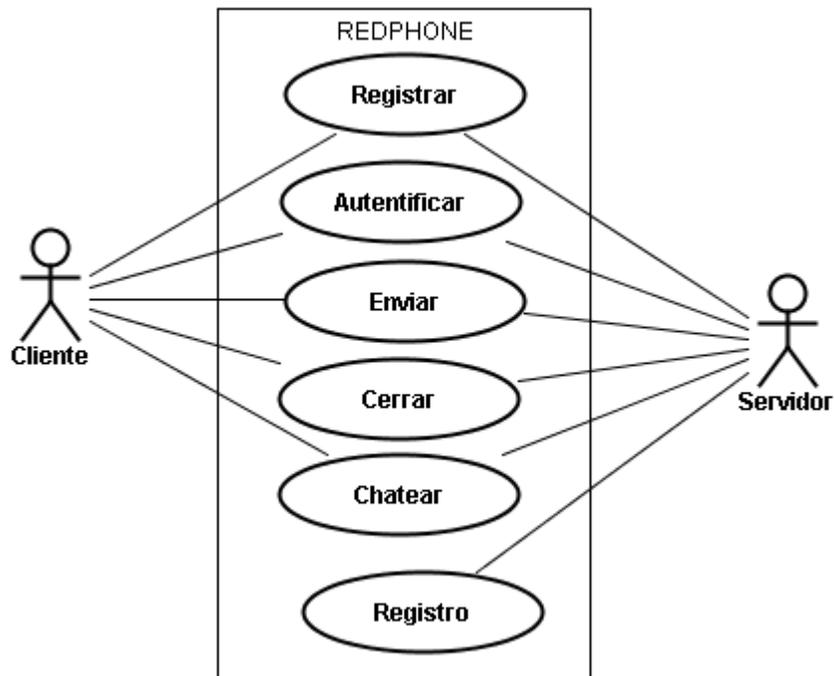


Figura 18. Diagrama de Casos de Uso

4.6.1.1 Especificación de casos de uso

Lista de especificaciones de casos de uso (CU):

Nombre Caso de Uso: CU1. REGISTRAR.	
Actores Principales:	Cliente, Servidor.
Resumen:	El usuario instala la aplicación en el dispositivo móvil, luego inicia la aplicación y en la pantalla de inicio escoge la opción registrarme.
Flujo de Eventos	
Camino Básico (CB):	<ol style="list-style-type: none"> 1. El usuario llena el formulario de registro en el cual ingresa sus datos personales. 2. El sistema verifica el Nick suministrado por el usuario. 3. El sistema verifica que los demás datos suministrados por el usuario cumplan los requisitos. 4. El sistema asigna al usuario a su respectiva red lógica, para determinar sus amigos. 5. El usuario es creado en la base de datos.
Camino Alternativo (CA):	<p>[A1] En el paso 2. Si el Nick que ingreso el usuario, está reservado por otro, se le notifica al usuario que no puede tener ese Nick.</p> <p>[A2] En el paso 3. Si los datos no cumplen con los requisitos mínimos, se le informa al usuario para que sean modificados.</p>
Postcondición:	<p>[CB] El sistema crea el nuevo usuario en la base de datos y lo envía a la interfaz de pantalla de inicio, para que el usuario pueda autenticarse e iniciar sesión.</p> <p>[CA] Se le informa al usuario por medio de una interfaz que no se puede registrar, de no modificar los datos suministrados.</p>

Tabla 8. Especificación CU1.

Nombre Caso de Uso: CU2. AUTENTICAR.	
Actores Principales:	Cliente, Servidor.
Resumen:	Haber iniciado la aplicación, estar en la pantalla de inicio de sesión y haberse registrado previamente.
Flujo de Eventos	
Camino Básico (CB):	<ol style="list-style-type: none"> 1. El usuario ingresa su respectivo Nick y contraseña, y selecciona la opción "OK". 2. El sistema envía los datos ingresados por el usuario y los compara

	<p>con la base de datos.</p> <p>3. El sistema verifica el resultado de la validación.</p> <p>4. El sistema carga la interfaz del panel de control.</p>
Camino Alternativo (CA):	[A1] En el paso 1. Si el usuario no se ha registrado, puede hacer clic en el botón registrarse para hacerlo (CU1).
	[A2] En el paso 3. Si los datos de Nick y contraseña no son correctos, se le informa al usuario para que los corrija.
Postcondición:	[CB] El sistema le envía los mensajes pendientes que ahí almacenados, el sistema informa que el usuario está en línea (Online).
	[CA] El sistema no continúa hasta que el Nick y contraseña sean correctos.

Tabla 9. Especificación CU2.

Nombre Caso de Uso: CU3. ENVIAR.

Actores Principales:	Cliente, Servidor.
Resumen:	Haberse autenticado correctamente el usuario y estar en la interfaz panel de control.
Flujo de Eventos	
Camino Básico (CB):	<ol style="list-style-type: none"> 1. El sistema está en espera a eventos generados por el usuario. 2. El usuario selecciona la opción "Enviar". 3. El sistema carga la interfaz de mensajes. 4. El sistema determina y muestra todos los amigos del usuario. 5. El usuario selecciona un amigo. 6. El usuario redacta un mensaje para el amigo seleccionado. 7. El usuario selecciona la opción "Enviar".
Camino Alternativo (CA):	<p>[A1] En el paso 1. Si el usuario quiere no quiere continuar puede cerrar sesión oprimiendo la opción "Salir".</p> <p>[A2] En el paso 4. Si es un usuario servidor el sistema también carga todas las redes, para enviar a todos los usuarios.</p> <p>[A3] En el paso 7. Si el usuario no quiere enviar el mensaje, puede seleccionar la opción "Atrás".</p>
Postcondición:	<p>[CB] El sistema determina si el usuario está en línea para enviarle el mensaje, si no está conectado lo almacena en temporales para enviárselo en su próxima conexión y guarda una copia en enviados.</p> <p>[CA] El sistema carga la interfaz de panel de control, y no envía el mensaje por petición del usuario.</p>

Tabla 10. Especificación CU3.

Nombre Caso de Uso: CU4. CERRAR.

Actores Principales:	Cliente, Servidor.
Resumen:	Haberse autenticado correctamente el usuario y estar en la interfaz panel de control.
Flujo de Eventos	
Camino Básico (CB):	<ol style="list-style-type: none"> 1. El sistema está en espera a eventos generados por el usuario. 2. El usuario ha decidido no utilizar más la aplicación y escoge la opción "Salir". 3. El sistema quita al usuario de los conectados (Online), el sistema cierra la sesión del usuario borrando los temporales. 4. El sistema carga la interfaz de pantalla de inicio. 5. El usuario escoge la opción "Cerrar".
Camino Alternativo (CA):	[A1] Si el usuario quiere continuar utilizando la aplicación no escoge la opción "Cerrar".
Postcondición:	<p>[CB] El sistema cierra la aplicación, permitiendo continuar con el uso del celular en otras funciones.</p> <p>[CA] El usuario puede volver a autenticarse de no oprimir la opción "Cerrar".</p>

Tabla 11. Especificación CU4.

Nombre Caso de Uso: CU5. CHATEAR.

Actores Principales:	Cliente, Servidor.
Resumen:	Haberse autenticado correctamente el usuario y estar en la interfaz panel de control.
Flujo de Eventos	
Camino Básico (CB):	<ol style="list-style-type: none"> 1. El usuario selecciona la opción "Conectados". 2. El sistema carga la interfaz donde muestra todos los usuarios en línea (Online) en una lista. 3. El usuario selecciona un usuario de la lista, luego escoge la opción chatear. 4. El sistema carga la interfaz para chatear y le informa al otro usuario que alguien quiere chatear.
Camino Alternativo (CA):	[A1] En el paso 4. Si el usuario esta como servidor, puede hacer eventos remotos a su respectivo cliente.
Postcondición:	<p>[CB] Los usuarios envían y reciben mensajería instantánea, y el sistema se encarga de informar los resultados.</p> <p>[CA] Si el usuario no quiere chatear más, puede cerrar seleccionando la opción "Cerrar".</p>

Tabla 12. Especificación CU5.

Nombre Caso de Uso:	CU14. REGISTROS.
Actores Principales:	Cliente, Servidor.
Resumen:	Cualquier evento o petición de un cliente hacia el servidor.
Flujo de Eventos	
Camino Básico (CB):	<ol style="list-style-type: none"> 1. El sistema está en estado de recepción donde espera las peticiones del cliente. 2. El sistema recibe y analiza la petición del cliente. 3. El sistema analiza y determina el tipo de petición clasificándola y organizándola. 4. El sistema filtra y crea registros para todas las peticiones.
Camino Alternativo (CA):	[A1] En el paso 4. Si la petición no pasa el filtrado, no se crea el registro en la base de datos.
Postcondición:	Crea lista de registros de las peticiones de los usuarios, almacenando la fecha, hora, hash del archivo o mensaje y Nick del usuario.

Tabla 13. Especificación CU14.

4.6.2 Pruebas

La arquitectura cliente servidor (C/S), representa un desafío para las pruebas del software, por los aspecto de rendimiento asociados proceso de las complejidades de las comunicaciones en red y la necesidad de servir a los clientes desde una base centralizada (Pressman, 2002). Por esto las pruebas que realizamos a continuación, son en un entorno controlado, con datos falsos y asegurando que la comunicación entre el cliente y el servidor este funcionando correctamente.

4.6.2.1 Caja Blanca

La prueba de caja blanca, denominada a veces prueba de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Para este caso se utiliza la prueba del camino básico (Pressman, 2002), este le permite al diseñador de casos obtener una medida de complejidad de un diseño procedimental y usar esta medida como guía.

Las pruebas de caja blanca realizadas a la aplicación Redphone, siguen la siguiente metodología. Android posee un registro de eventos ocurridos donde se puede observar con detalle todo lo ocurrido en el móvil, este registro se denomina logcat, esto es muy útil para detectar los posibles fallos del aplicativo que se ejecuta en el móvil o errores de desarrollo de la aplicación. Para poder observar estos logs podemos utilizar la herramienta denominada ddms que trae el SDK de Android (La ruta de este ejecutable es: AccesoDirecto/tools/ddms.bat), cuando se ejecuta este archivo se puede observar la siguiente ventana:

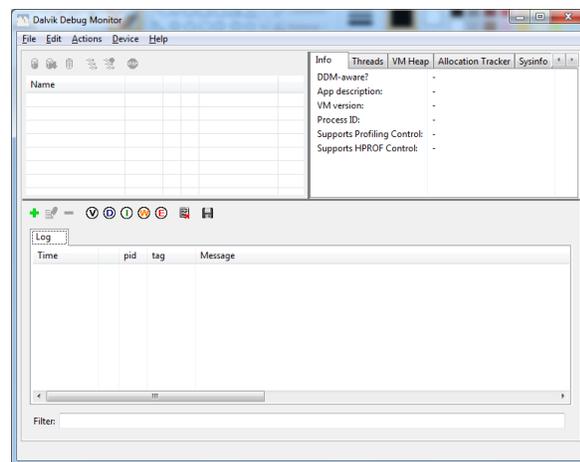


Figura 19. Prueba Caja Blanca – DDMS

En los botones superiores al registro se puede hacer un filtro de mensajes según su género:

- [E] Error: Muestra solo errores graves del sistema. (en rojo)
- [W] Warning: Muestra advertencias, errores leves o del algún tipo. (en naranja)
- [I] Information: Muestra mensajes de información. (en verde)
- [D] Debug: Muestra mensajes de depuración. (en azul)
- [V] Verbose: Muestra todos los mensajes.

Logrando de esta manera poder detectar errores lógicos de programación, utilizando principalmente dos filtros que son: Error y Debug.

4.6.2.2 Caja Negra

Las pruebas de caja negra también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software (Pressman, 2002). Con estas pruebas se puede llegar a obtener errores de las siguientes categorías:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores de sentencias, estructura de datos o acceso a bases de datos.
4. Errores de rendimiento.
5. Errores de inicialización o terminación.

Las siguientes tablas representan los resultados obtenidos de la aplicación RedPhone v3.0.

Nombre Prueba:	Registrar Usuario
Precondición:	El actor (Cliente) esté conectado con el actor (Servidor). Y ambos hayan iniciado la aplicación.
Descripción:	En esta prueba se realiza con ambos actores paralelamente, iniciamos con un actor y en la pantalla de inicio escogemos la opción "registrar". Aparece un formulario y procedemos a llenar los datos.
Entradas:	Nombre = "Jimena", Apellidos = "Ruiz", Nick = "Jimena", Edad = "21", Contraseña = "1234", Repita Contraseña = "1234", En la opción carrera aparece un lista y seleccionamos la carrera = "Redes y Seguridad".
Resultado Esperado:	El usuario se ha creado correctamente. Luego procede a cargar la interfaz de pantalla de inicio.
Evaluación:	Prueba superada.

Tabla 14. Resultados Pruebas P1.

Nombre Prueba:	Autenticar Usuario
Precondición:	El actor (Cliente) esté conectado con el actor (Servidor). Y tenga creado un usuario previamente en la base de datos y proceder a identificarnos con ese usuario.
Descripción:	En esta prueba se realiza con ambos actores paralelamente, iniciamos con un actor y en la pantalla de inicio ingresamos en respectivo usuario y contraseña.
Entradas:	Nick = "Andreita", Pass = "1234".
Resultado Esperado:	Cargar la interfaz del panel de control, con los datos personales correspondientes e indicar si hay nuevos mensajes en la bandeja de entrada.
Evaluación:	Prueba superada.

Tabla 15. Resultados Pruebas P2.

Nombre Prueba:	Enviar Mensajes
Precondición:	El actor (Cliente) esté conectado con el actor (Servidor). Tener creado mínimo dos usuarios previamente en la base de datos para enviar el mensaje de uno a otro. Y haber iniciado sesión con un actor.
Descripción:	En esta prueba se realiza con ambos actores paralelamente, iniciamos con un actor y en la interfaz panel de control seleccionamos la opción "Enviar", carga la nueva interfaz, luego en la lista de amigos seleccionamos al cual queremos enviar, redactamos el mensaje y de nuevo escogemos la opción "Enviar".
Entradas:	Para = "Daniela", Mensaje = "Hola este es el mensaje de prueba."
Resultado Esperado:	Volver a la interfaz del panel de control y mostrar un mensaje de "Enviado Correctamente".
Evaluación:	Prueba superada.

Tabla 16. Resultados Pruebas P3.

Nombre Prueba:	Chatear
Precondición:	El actor (Cliente) esté conectado con el actor (Servidor). Tener creado mínimo dos usuarios previamente en la base de. Y haber iniciado sesión con los dos usuarios que van a chatear, uno con cada actor.
Descripción:	En esta prueba se realiza con ambos actores paralelamente, iniciamos sesión con cada actor, pero con usuarios diferentes y en la interfaz panel de control de un actor seleccionamos la opción "Chatear", carga la nueva interfaz, luego en la lista de conectados seleccionamos con cual vamos a chatear, luego escogemos la opción "Chatear", esto le informa al otro usuario que alguien desea chatear con él, cuando el otro usuario acepta, se carga la interfaz de chat en cada acto y empieza la mensajería. Cada uno redacta su mensaje y oprime la opción "Enviar".
Entradas:	Mensaje = "hola".
Resultado Esperado:	Que el mensaje escrito por un actor sea recibido por el otro, limpiar la entrada y mostrar el historial de los mensajes.
Evaluación:	Prueba superada.

Tabla 17. Resultados Pruebas P4.

Capitulo V: CRONOGRAMA

Los que emplean mal su tiempo son los primeros en quejarse de su brevedad. (Jean De La Bruyere)

5. CRONOGRAMA

5.1 CRONOGRAMA DE ACTIVIDADES

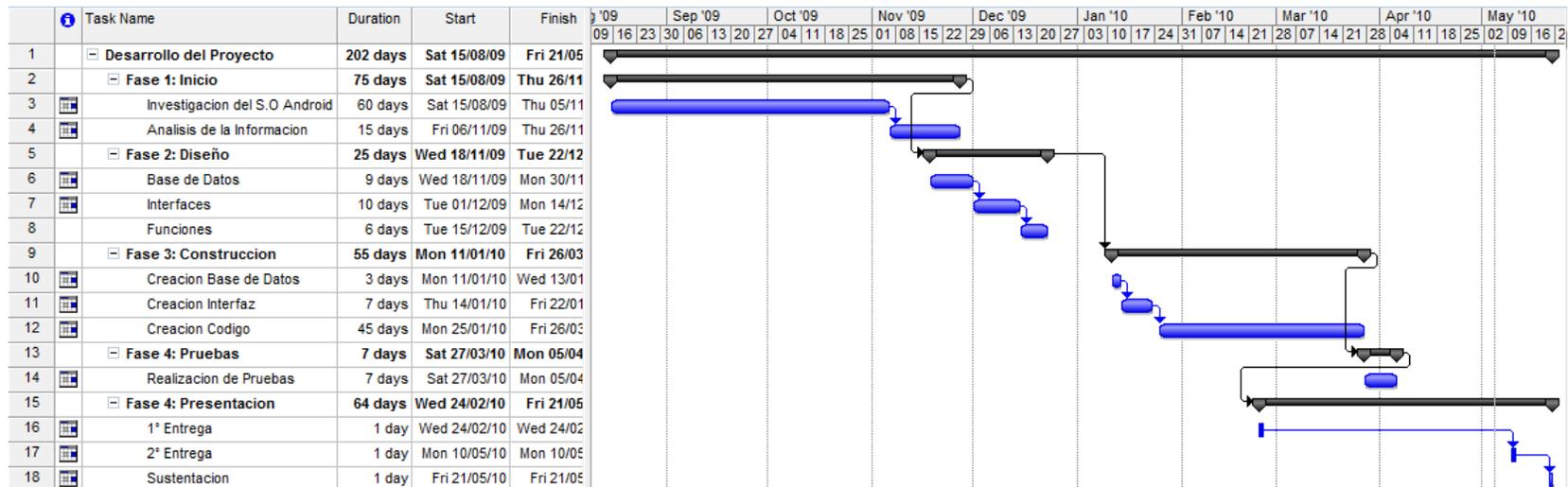


Tabla 18. Actividades.

5.2 PRESUPUESTO

Recursos Humanos				
Elemento	Descripción	Horas	Valor Hora	Total
Técnico Profesional en Sistemas	Analista y Creador de Código	480	\$ 12.000	\$ 5.760.000
				\$ -
		Total:		\$ 5.760.000

Recursos Técnicos				
Elemento	Descripción	Cantidad	Costo Unidad	Total
Sistema Operativo	Linux Ubuntu 9.10	1	\$ -	\$ -
OpenOffice	3.0	1	\$ -	\$ -
NeatBeans	IDE de Java	1	\$ -	\$ -
SDK	AVD Simulador de Dispositivo Movil Android	1	\$ -	\$ -
PC	Turion 64x2, 2 GB RAM, 160 GB DD	1	\$ 1.700.000	\$ 1.700.000
Impresora	HP PhotoSmart C4280	1	\$ 325.000	\$ 325.000
		Total:	\$	2.025.000

COSTO TOTAL DEL PROYECTO	\$ 7.785.000,00
---------------------------------	------------------------

Tabla 19. Presupuesto.

Capitulo VI: CONCLUSIONES

La vida es el arte de sacar conclusiones suficientes a partir de datos insuficientes. (Samuel Butler)

6. CONCLUSIONES

Con Android se está viendo el proceso evolutivo de un sistema operativo, desde sus inicios en el año 2007, no aparentaba ser un sistema que fuera digno de competencia para las grandes empresas que manejan el mercado de los celulares móviles, sin embargo y tan solo transcurridos dos años de su lanzamiento con el respaldo del magnate Google, Android se ha posicionado globalmente abarcando el mercado europeo y americano, llamando la atención a empresas como Motorola, Samsung, HTC, entre otras, para lograr que estas incorporen dicho sistema en sus productos para satisfacer las necesidades de los usuarios (Pareciera que todos pelean por tener su terminal con Android).

Por ser Android de código abierto y basado sobre kernel de Linux, se puede afirmar que nunca le faltaran programadores ni aplicativos que satisfagan las necesidades del mercado, todo esto gracias a la ya bastante grande comunidad que ha formado Android, permitiendo mostrar aplicativos en el “Android Market” a cualquier usuario programador que este dispuesto a enviarle su producto, cosa que motiva mas a los programadores que no pertenecen a las empresas, algo que no hacen ni Microsoft ni Apple OS.

Android no solo fue diseñado para celulares, y si sigue creciendo su popularidad a este ritmo, dentro de muy poco se podrá verlo en todo tipo de aparatos móviles (cámaras, mp3, mp4, etc.), se podría decir que Android se convertirá en el *Windows de los móviles*.

RedPhone es una muestra de que con el sistema Android y una buena idea, se pueden lograr muy buenos aplicativos, ya que con las API's que maneja le permite a un usuario programador desarrollar eventos, interfaces, acciones, etc., con pocas líneas de código y fáciles de comprender.

6.1 RECOMENDACIONES

Si se cuenta con la fortuna de tener un dispositivo Android a la mano o no, hay que saber que existen varias grandes comunidades, es bueno conocerlas hay que visitarlas y encontrar una que logre satisfacer las necesidades, por lo general estas se dedican a crear aplicaciones, foros de dudas, manuales, ROMs, etc, hacen modificaciones al sistema para lograr personalizar a una mayor escala cada dispositivo, acá se muestran algunas que ayudaron al desarrollo de este proyecto.

- ✓ XDA
- ✓ Modaco
- ✓ HTCMania
- ✓ Android-Spa

Las comunidades en general, no necesariamente las de Android, son creadas bajo filosofía de ayudar a los demás, hay un dicho: "Alguien ya pensó y hizo lo que ud tiene que hacer", por lo cual no somos los primeros ni los últimos para desarrollar y encontrar problemas de todo tipo. Y no son solo para resolver problemas, también fomentan la integración y la ayuda desinteresada, lo que hace reflexionar y mejorar personalmente.

7. BIBLIOGRAFÍA

Libros:

Pressman, R. (2002). Ingeniería del Software. Mc Graw Hill.

Articulos web:

Android. (s.f.). Open Movil Forum. Recuperado el Octubre de 2009, de <http://open.movilforum.com/blog/introduccion-android>

TIE. (s.f.). Universidad de Medellin. Recuperado el Octubre de 2009, de <http://www.udem.edu.co/UDEM/Agenda/Externos/android.htm>

Comunidad Android-Spa: Usuario: "Tavo88"

<http://www.android-spa.com/>

Figuras tomadas de:

Figura 1. http://kronox.org/imagenes/diagrama_android.png

Figura 2. http://kronox.org/imagenes/estructura_android.png

Figura 3. <http://www.androphones.com/all-android-phones.jpg>

A N E X O S: